

Hướng Dẫn Thực Hành

LINQ

Đối tượng: Tất cả



GV Biên Soạn: **Ths.Trần Duy Thanh**

Phone: **0987773061**

Email: duythanhcse@gmail.com

Blog: <http://duythanhcse.wordpress.com/>

Facebook: <https://www.facebook.com/duythanhcse>

Module 1: Truy suất dữ liệu với LINQ

Nội dung kiến thức thực hành:

- Thực hành LINQ:
- Hiểu được Generic
- Hiểu được Implicitly Typed Variables
- Hiểu được Anonymous Types
- Hiểu được Extension Methods
- Hiểu được Lambda Expressions
- Hiểu được LinQ to Object
- Hiểu được LinQ to SQL
- Sinh viên có thể tải toàn bộ ví dụ liên quan tới LINQ từ trang chính thống của Microsoft để tham khảo: <http://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b>

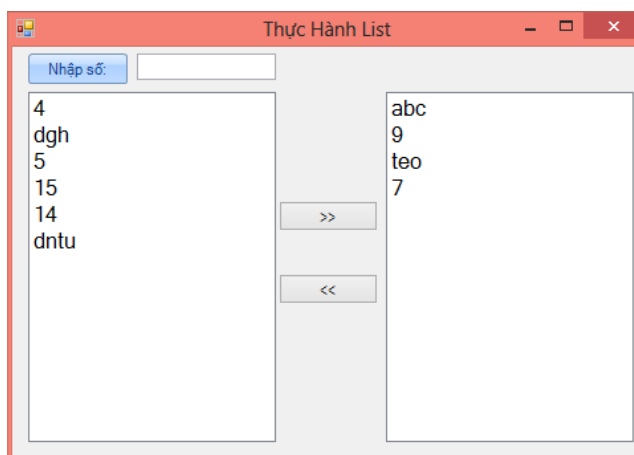
Bài tập 1:

Mục đích:

- Hiểu được Generic
- Trong **System.Collections.Generic**; cung cấp rất nhiều class: List, Dictionary, SortedList, SortedDictionary, LinkedList, HashSet....
- Nhưng ở đây chúng ta chỉ quan tâm tới **List** và **Dictionary**, các class khác sinh viên phải có trách nhiệm tự nghiên cứu.

Yêu cầu:

- Sử dụng **List** để viết chương trình như bên dưới:



- Mỗi lần click “Nhập số”: Sẽ đưa số trong TextBox vào Listbox bên trái.
- Click “>>” sẽ chuyển tất cả các phần tử đang được chọn trong ListBox bên trái sang ListBox bên phải

- Click “<<” sẽ chuyển tất cả các phần tử đang được chọn trong ListBox bên phải sang Listbox bên trái
- Cải tiến lại chương trình: cho phép đưa bất kỳ kiểu dữ liệu nào vào Listbox (xem hình)

Hướng dẫn:

- Thiết lập các ListBox có chế độ cho phép chọn nhiều phần tử: SelectionMode là **MultiSimple** hoặc **MultiExtended**.
- Để List có thể chứa bất kỳ kiểu dữ liệu nào thì ta dùng List<**object**>. Mọi class sinh ra đều kế thừa từ object, nên khi ta để object thì nó sẽ có thể lưu bất kỳ kiểu dữ liệu nào.

Bài tập 2:**Mục đích:**

- Hiểu được Generic
- Trong **System.Collections.Generic**; cung cấp rất nhiều class: List, Dictionary, SortedList, SortedDictionary, LinkedList, HashSet....
- Nhưng ở đây chúng ta chỉ quan tâm tới **List** và **Dictionary**, các class khác sinh viên phải có trách nhiệm tự nghiên cứu.

Yêu cầu:

- Sử dụng **Dictionary** để viết chương trình như bên dưới:

STT	Mã sinh viên	Tên sinh viên
1	010103	Nguyễn Văn Tèo
2	020103	Trần Thị Tí
3	040302	Nguyễn Ngọc An

Thông tin chi tiết

Mã sinh viên:

Tên sinh viên:

- Mỗi lần bấm Lưu: Đưa thông tin sinh viên vào Dictionary rồi cập nhật lên ListView. Nếu mã đã tồn tại thì tự động cập nhật, nếu mã chưa tồn tại thì thêm mới.
- Mỗi lần click vào từng phần tử trong ListView thì hiển thị thông tin chi tiết của sinh viên vào phần Thông tin chi tiết
- Nút Xóa: cho phép xóa sinh viên hiện tại
- Chú ý là tất cả các thao tác phải sử dụng Dictionary.
- Khi thao tác với CSDL thì Dictionary là một class rất hữu hiệu, cải tiến tốc độ xử lý.

Hướng dẫn:

- Tạo một class tên là Sinhvien
- Khai báo Dictionary như sau:
`Dictionary<string, sinhvien> dic = new Dictionary<string, sinhvien>();`
 Với đối số thứ nhất là key, đối số thứ 2 là value
- Ta có thể đưa dữ liệu vào bằng phương thức `dic.Add("010203", sinhvien nào đó)`
- Để duyệt toàn bộ các phần tử trong Dictionary có nhiều cách, ở đây ta thường dùng:

```
foreach (KeyValuePair<string, sinhvien> item in dic)
{
    sinhvien sv = item.Value;
}
```

- Để lấy sinh viên theo đúng mã ta chỉ cần: `dic["010203"]`
- Để xóa sinh viên ta làm như sau: `dic.Remove("010203");`

Bài tập 3:**Mục đích:**

- Thực hành và hiểu được **Implicitly Typed Variables**

Yêu cầu:

- Hãy khai báo và gán giá trị bất kỳ cho một biến có kiểu **var**

```
var x = 113;
var y = "1/1/2012";
var z = 1.7;
var k = new DateTime(2012, 1, 1);
string msg = "x type="+x.GetType() + "\n"+
            "y type ="+y.GetType() + "\n" +
            "z type ="+z.GetType() + "\n" +
            "k type = " + k.GetType();
MessageBox.Show(msg);
```

- Hãy thử trường hợp không gán giá trị mặc định cho biến có kiểu **var**, cho nhận xét
- Hãy thử gán giá trị có kiểu dữ liệu khác cho biến đã khai báo kiểu **var**, cho nhận xét
- Hãy cho biết **var** thường được ứng dụng trong trường hợp nào?
- Trong trường hợp đã biết chính xác kiểu dữ liệu thì có nên khai báo kiểu **var** hay không?

Hướng dẫn:

- **var** được .net framework hỗ trợ cơ chế nội suy kiểu dữ liệu
- do đó nó sẽ tự động lấy đúng kiểu dữ liệu khi ta gán vào cho nó.
- Bất kỳ kiểu dữ liệu nào, kể cả kiểu object, var cũng tự động nội suy ra đúng kiểu dữ liệu của nó:
- **var s=new sinhvien();** thì s cũng được nội suy ra là đối tượng sinh viên, do đó ta có thể sử dụng các Properties, method .. của sinhvien một Cách bình thường.

Bài tập 4:

Mục đích:

- Thực hành và hiểu được **Anonymous Types**

Yêu cầu:

- Hãy kiểm tra một số đoạn lệnh sau:

```
var teo = new { ID=1234, Name="Tèo Hả Tèo"};  
MessageBox.Show(teo.ID + "-" + teo.Name);
```

- Sinh viên tự tạo thêm nhiều Anonymous Types để hiểu thêm về nó.
- Anonymous Types được sử dụng trong nhiều tình huống khác nhau.

Hướng dẫn:

- Anonymous Types được sử dụng khi:
 - ✓ Ta cần một đối tượng tạm thời để lưu trữ dữ liệu
 - ✓ Khi ta không cần định nghĩa phương thức
 - ✓ Khi chúng ta muốn tạo thêm các Properties khác ngoài class đã định nghĩa
 - ✓ Khi ta muốn thay đổi thứ tự các properties...

Bài tập 5:

Mục đích:

- Thực hành và hiểu được Extension Methods:
 - ✓ Quy tắc tạo extension methods như thế nào?
 - ✓ Cách sử dụng chúng ra sao?

Yêu cầu:

- Hãy cài một hàm tính tổng các số từ 1 tới N vào kiểu số nguyên
- Hãy cài một hàm kiểm tra số nguyên tố vào kiểu số nguyên
- Hãy cài một hàm xuất danh sách các số nguyên tố vào kiểu số nguyên
- Hãy cài một hàm xuất danh sách dãy số Fibonacci vào kiểu số nguyên
- Hãy cài một hàm cho phép nối 2 chuỗi vào kiểu chuỗi
- Hãy cài một hàm cho phép tô màu đỏ vào Button
- Hãy cài một hàm cho phép tô đen các số chẵn vào ListBox
- Hãy cài một hàm cho phép tô đen các số lẻ vào ListBox
- Hãy cài một hàm cho phép tô đen các số nguyên tố vào ListBox
- Dùng delegate làm mặt nạ để thực hiện tô số chẵn, tô số lẻ, tô số nguyên tố cho ListBox

Hướng dẫn:

```
namespace StudyLINQ  
{ public static class MyExtensionMethod  
{  
    public static int SumFrom1toN(this int n)  
    {  
        int sum = 0;  
        for (int i = 1; i <= n; i++)  
            sum += i;  
        return sum;  
    }  
}
```

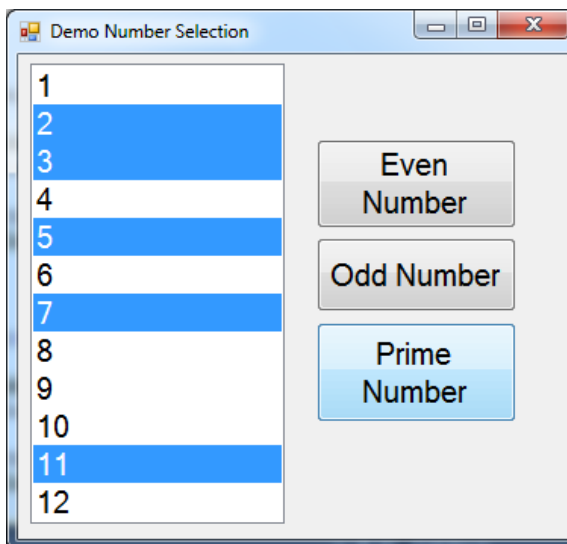
Bài tập 6:

Mục đích:

- Thực hành và hiểu được Lambda Expressions

Yêu cầu:

- Sử dụng Lambda Expressions để làm bài tập như hình dưới đây:



- Thực hiện 3 chức năng:
 - ✓ Tô đen số chẵn
 - ✓ Tô đen số lẻ
 - ✓ Tô đen số nguyên tố

Hướng dẫn:

- Kết hợp Extension methods với Lambda Expression để làm bài tập này:

`Listbox1.todensochan(x => x % 2 == 0);`

- Với hàm **todensochan**, viết theo kỹ thuật Extension methods

```
public static class MyExt
{
    public static void todensochan(this ListBox lb)
    {
        //viết code tô đen ở đây
    }
}
```

Bài tập 7:**Mục đích:**

- Thực hành và hiểu được Query Syntax và Method Syntax
- So sánh được Query Syntax với Method Syntax
- Hiểu được **Deferred execution** và **lazy loading**

Yêu cầu:

- Sinh viên hãy thực hành lại các đoạn lệnh bên dưới đây:
- **Lệnh số 1:**

```
string[] list = new string[] { "teo", "ty", "bin", "bo"};
```

```
var ret1 = from c in list
           where c.StartsWith("t")
           select c;
```

```
IEnumerable<string> ret2 =
    list.Where(c=>c.StartsWith("t"));
```

- **Lệnh số 2:**

```
string[] list = new string[] { "teo", "ty", "bin", "bo"};
```

```
var ret1 = from c in list
           where c.EndsWith("o")
           orderby c
           select new { Id=c.ElementAt(0),Name=c };
```

```
var ret2 = list
    .Where(c => c.EndsWith("o"))
    .OrderBy(c => c)
    .Select(c=>new { Id=c.ElementAt(0),Name=c});
```

- **Lệnh số 3:**

```
var source = new List<string> { "A", "B", "C"};
var values = from c in source
             select c;
source.Add("D");
foreach (var c in values)
{
    Console.WriteLine(c);
}
```

Out Put:

A
B
C
D

Hướng dẫn:

- Tương tự như câu truy vấn SQL, Query Syntax cũng dùng quy tắc tương tự nhưng ở đây nó đảo ngược from lên trước. Ở đâu thấy from ... select đó chính là query syntax
- Lambda expression thường được sử dụng trong Method Syntax

Bài tập 8:**Mục đích:**

- Thực hành và hiểu được LinQ to Object, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **ForEach, Exists, TrueForAll**
 - ✓ Thực hành và hiểu được **Find, FindAll, FindIndex, FindLast, FindLastIndex**
 - ✓ Thực hành và hiểu được **RemoveAll**
 - ✓ Và các hàm khác...
- Hiểu được Generic **List**

Yêu cầu:

- Một sản phẩm cần có các thông tin (mã sản phẩm, tên sản phẩm, số lượng, đơn giá, xuất xứ, ngày hết hạn)
- Hãy viết chương trình quản lý sản phẩm đáp ứng các yêu cầu sau:
 - ✓ Cho phép thêm/ sửa/ xóa sản phẩm
 - ✓ Cho phép duyệt danh sách sản phẩm
 - ✓ Kiểm tra xem trong kho có chứa bất kỳ sản phẩm nào quá hạn hay không?
 - ✓ Tìm 1 sản phẩm có đơn giá cao nhất
 - ✓ Tìm 1 sản phẩm có xuất xứ từ Nhật Bản
 - ✓ Xuất tất cả các sản phẩm bị quá hạn trong kho
 - ✓ Xuất tất cả các sản phẩm có đơn giá trong đoạn [a...b]
 - ✓ Xóa các sản phẩm có xuất xứ bất kỳ
 - ✓ Xóa toàn bộ sản phẩm trong kho
- Giao diện tương tự như bên dưới:

The screenshot shows a Windows application window titled "LINQ to OBJECT - Quản lý sản phẩm". It contains several functional areas:

- Input Section (Left):** Labeled "Nhập thông tin sản phẩm:", it includes text boxes for "Mã SP:", "Tên SP:", "Số lượng:", "Đơn giá:", "Xuất xứ:", and "Ngày hết hạn:". Below these are "Lưu SP" and "Xóa SP" buttons.
- Search Section (Top Right):** Labeled "Chọn chức năng tìm kiếm:", it features buttons for "1 SP có ĐG cao nhất", "1 SP từ Nhật Bản", "Xuất toàn bộ SP quá hạn", and "Xuất các SP có ĐG [a...b]:". The last button has input fields for the values "500" and "800".
- Product List Table (Middle Right):** A table with columns: Mã SP, Tên SP, Số lượng, Đơn giá, Xuất xứ, and Ngày hết hạn. It contains two rows of data:

Mã SP	Tên SP	Số lượng	Đơn giá	Xuất xứ	Ngày hết hạn
sp2	Kem Chiên	12	800	Lào	12/5/2013
sp3	Cá viên nướng	100	500	Việt Nam	30/7/2013
- Product List After Input (Bottom Left):** Labeled "Danh sách sản phẩm sau khi nhập:", it shows a table with the same columns as above, containing three rows:

Mã SP	Tên SP	Số lượng	Đơn giá	Xuất xứ	Ngày hết hạn
sp1	Sữa Chua Vinamilk	10	15000	Trung Qu...	15/8/2013
sp2	Kem Chiên	12	800	Lào	12/5/2013
sp3	Cá viên nướng	100	500	Việt Nam	30/7/2013
- Action Section (Bottom Right):** Labeled "Chọn thao tác:", it includes buttons for "Xóa SP theo xuất xứ bất kỳ:", "Kiểm tra kho có SP quá hạn hay không?", "Xóa toàn bộ SP trong kho", and "Xóa toàn bộ SP quá hạn".

Hướng dẫn:

- Sử dụng Generic List để thao tác với toàn bộ yêu cầu đặt ra.

Bài tập 9:**Mục đích:**

- Thực hành và hiểu được LinQ to Object, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **Select**
 - ✓ Thực hành và hiểu được **Where**
 - ✓ Thực hành và hiểu được **OrderBy/OrderByDescending**
 - ✓ Thực hành và hiểu được **All, Any, Max, Min**
 - ✓ Sử dụng **Metro Form** và một số control khác của **DotnetBar**

Yêu cầu:

- Viết chương trình cho phép nhập vào một danh sách N số ngẫu nhiên có giá trị từ -500 tới 500 rồi thực hiện các tác vụ giống như giao diện dưới đây:

The screenshot shows a Windows application titled "LinQ to Object". It has three main panels:

- Nhập số ngẫu nhiên:** A panel with a "Nhập" (Input) text box containing the number "12" and a list of 12 random numbers: 342, 165, -26, 247, -127, 320, 219, 103, 392, -201, -307, 495.
- Chọn tác vụ:** A panel with buttons for various LINQ operations: Select, Where, OrderBy (which is currently selected), OrderByDescending, All, Any, Max, and Min.
- Kết quả:** A panel displaying the result of the selected operation. Since "OrderBy" is selected, the numbers are sorted in ascending order: -307, -201, -127, -26, 103, 165, 219, 247, 320, 342, 392, 495.

- ✓ **Select:** Lấy ra số hàng đơn vị của mỗi số trong danh sách
- ✓ **Where:** Trích ra danh sách các số nguyên tố trong danh sách
- ✓ **OrderBy:** Sắp xếp danh sách tăng dần
- ✓ **OrderyDescending:** Sắp xếp danh sách giảm dần
- ✓ **All:** Kiểm tra xem có phải tất cả các phần tử trong danh sách là số âm hay không?
- ✓ **Any:** Kiểm tra xem danh sách có bất kỳ một phần tử nào là số hoàn thiện hay không? (số hoàn thiện là số có tổng các ước số không kể nó thì bằng chính nó, ví dụ $6=1+2+3$, $28=1+2+4+7+14$)
- ✓ **Max:** xuất ra số lớn nhất trong danh sách
- ✓ **Min:** xuất ra số nhỏ nhất trong danh sách

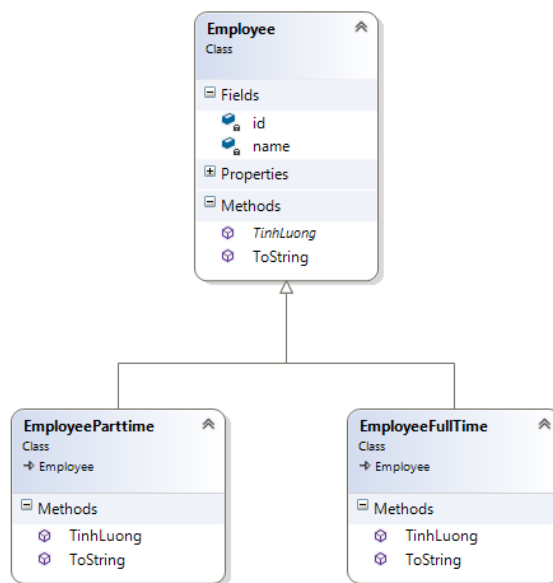
Hướng dẫn:

Bài tập 10:**Mục đích:**

- Thực hành và hiểu được LinQ to Object, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **Select, Where**
 - ✓ Thực hành và hiểu được **OrderBy/OrderByDescending**
 - ✓ Thực hành và hiểu được **OfType, All, Any, Max, Min**
 - ✓ Và các hàm khác...

Yêu cầu:

- Viết chương trình quản lý nhân viên theo mô hình class dưới đây:



- Giao diện chương trình:

The screenshot shows the MetroForm application interface. It is divided into two main sections: "Danh sách nhân viên:" (Employee List) and "Nhập thông tin nhân viên:" (Enter employee information).

Danh sách nhân viên: This section contains a table with columns STT, Mã NV, Tên NV, and Lương. It also has three radio buttons for filtering: "Xem hết" (selected), "Chính thức", and "Thời vụ".

STT	Mã NV	Tên NV	Lương
1	NV01	Ái Nương	500
2	NV02	Hoàng Dung	400
3	NV03	Quách Tĩnh	1000

Nhập thông tin nhân viên: This section contains input fields for "Mã NV:", "Tên NV:", and "Ngày công:". There is also a checkbox for "Là thời vụ". Below these fields are "Lưu" (Save) and "Xóa" (Delete) buttons.

Chọn tác vụ: This section contains four buttons for different actions: "Chỉ duyệt theo tên NV", "Sắp NV theo Tên tăng dần", "Sắp NV theo lương giảm dần", and "Lọc NV có lương từ 300-> 500".

- Có 2 loại nhân viên: Nhân viên chính thức và nhân viên thời vụ
- Nhân viên chính thức thì mức lương là 1000.
- Nhân viên thời vụ thì dựa vào số ngày công *30
- Thực hiện các chức năng:
 - ✓ Người sử dụng checked vào “**Là thời vụ**” thì mới hiển thị mục ngày công, nếu không checked thì ẩn mục ngày công, mặc định khi khởi động chương trình thì unchecked “**Là thời vụ**”
 - ✓ Bấm nút “Lưu”, tự động lưu mới khi mã chưa tồn tại, tự động cập nhập khi mã đã tồn tại. Khi lưu thành công thì cập nhập vào ListView
 - ✓ Bấm nút “Xóa”, cho phép xóa nhân viên hiện tại đang chọn
 - ✓ Chỉ duyệt theo tên nhân viên: Phần ListViewEx sẽ chỉ có 2 cột là cột Thứ Tự và cột Tên nhân viên
 - ✓ Thực hiện “sắp nhân viên theo tên tăng dần” → cập nhập lại ListViewEx
 - ✓ Thực hiện “sắp nhân viên theo lương giảm dần” → cập nhập lại ListViewEx
 - ✓ Thực hiện lọc nhân viên có lương từ 300 tới 500 → cập nhập lại ListViewEx
 - ✓ Trong mục xem danh sách nhân viên, chọn loại nhân viên nào thì lọc các nhân viên đó vào ListViewEx

Hướng dẫn:

- Tất cả phải sử dụng **Generic List** và các phương thức của **List**
- Dùng tính đa hình trong hướng đối tượng để xử lý coding thật khoa học

Bài tập 11:

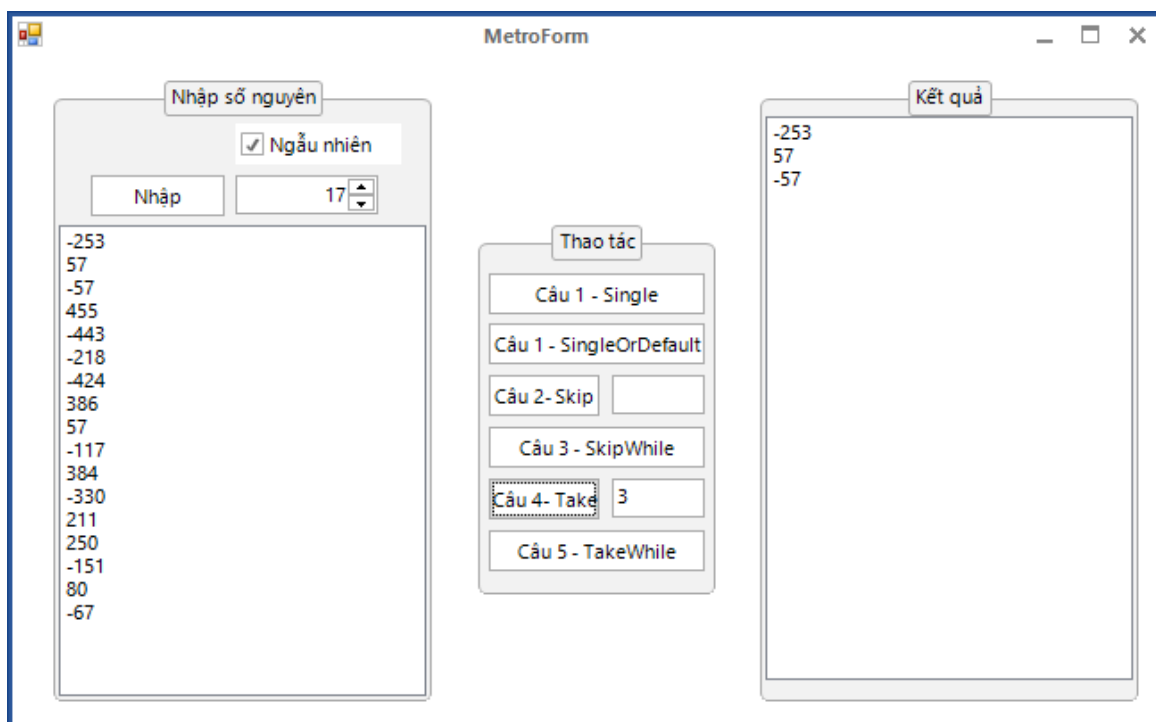
Mục đích:

- Thực hành và hiểu được LinQ to Object, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **Single, SingleOrDefault**
 - ✓ Thực hành và hiểu được **Skip, SkipWhile**
 - ✓ Thực hành và hiểu được **Take, TakeWhile**
 - ✓ Và các hàm khác...

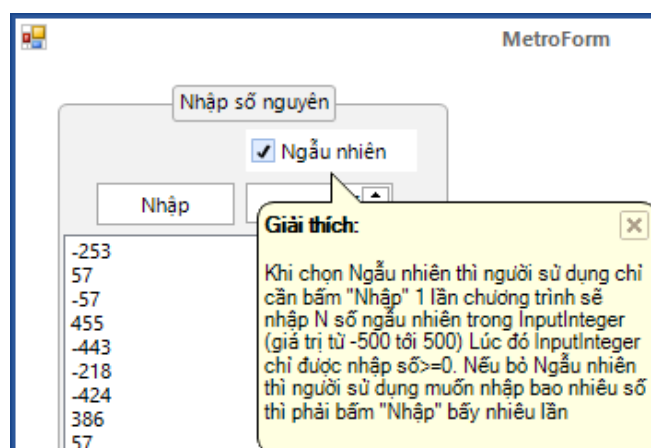
Yêu cầu:

- Viết chương trình cho phép nhập vào một danh sách các số nguyên rồi thực hiện các yêu cầu sau:
 - ✓ Lấy ra một số hoàn thiện trong danh sách, yêu cầu dùng hàm **Single** và **SingleOrDefault** (hãy cho biết chuyện gì xảy ra nếu danh sách có từ 2 số hoàn thiện trở lên)
 - ✓ Viết hàm cho phép bỏ qua **n** phần tử đầu tiên trong danh sách (dùng **Skip**)
 - ✓ Viết hàm cho phép loại bỏ các phần tử là số âm liên tiếp đầu tiên trong danh sách (dùng **SkipWhile**)
 - ✓ Viết hàm cho phép trích ra **n** phần tử đầu tiên trong danh sách (dùng **Take**)
 - ✓ Viết hàm cho phép trích ra các phần tử lớn hơn 50 liên tiếp đầu tiên trong danh sách (Dùng **TakeWhile**)

- Giao diện tương tự như hình dưới đây:



- Dùng **BalloonTip** để gắn cho checkbox Ngẫu nhiên:



Hướng dẫn:

- Tất cả phải sử dụng **Generic List** và các phương thức của **List**

Bài tập 12:

Mục đích:

- Thực hành và hiểu được LinQ to Object, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **ToArray**
 - ✓ Thực hành và hiểu được **ToList**
 - ✓ Thực hành và hiểu được **ToDictionary**
 - ✓ Và các hàm khác...

Yêu cầu:

- Thực hành lại các lệnh sau:
- **Kiểu int:**

```
List<Int32> myList = new List<int>();
myList.AddRange(new int[] { 2, 6, 7, 8, 0, 4, 9, 7 });
var myenumrable = myList.Where(x => x % 2 != 0);
int[] arr1 = myenumrable.ToArray();
List<int> list2 = myenumrable.ToList();
```

- **Kiểu Object:**

```
public class Person
{
    public string Id { get; set; }
    public string Name { get; set; }
}

List<Person> listPerson = new List<Person>();
listPerson.Add(new Person() { Id="1",Name="Tèo" });
listPerson.Add(new Person() { Id = "2", Name = "Hùng" });
listPerson.Add(new Person() { Id = "3", Name = "Bin" });
//Lấy Id làm key, Person làm value
Dictionary<string,Person> dic= listPerson.ToDictionary(x => x.Id);
foreach (KeyValuePair<string, Person> item in dic)
{
    string Id = item.Key;
    Person p = item.Value;
    //Xử lý Id, p
}

foreach (Person p in dic.Values)
{
    //Xử lý từng p
}

foreach (string id in dic.Keys)
{
    //xử lý từng id
}
```

Hướng dẫn:

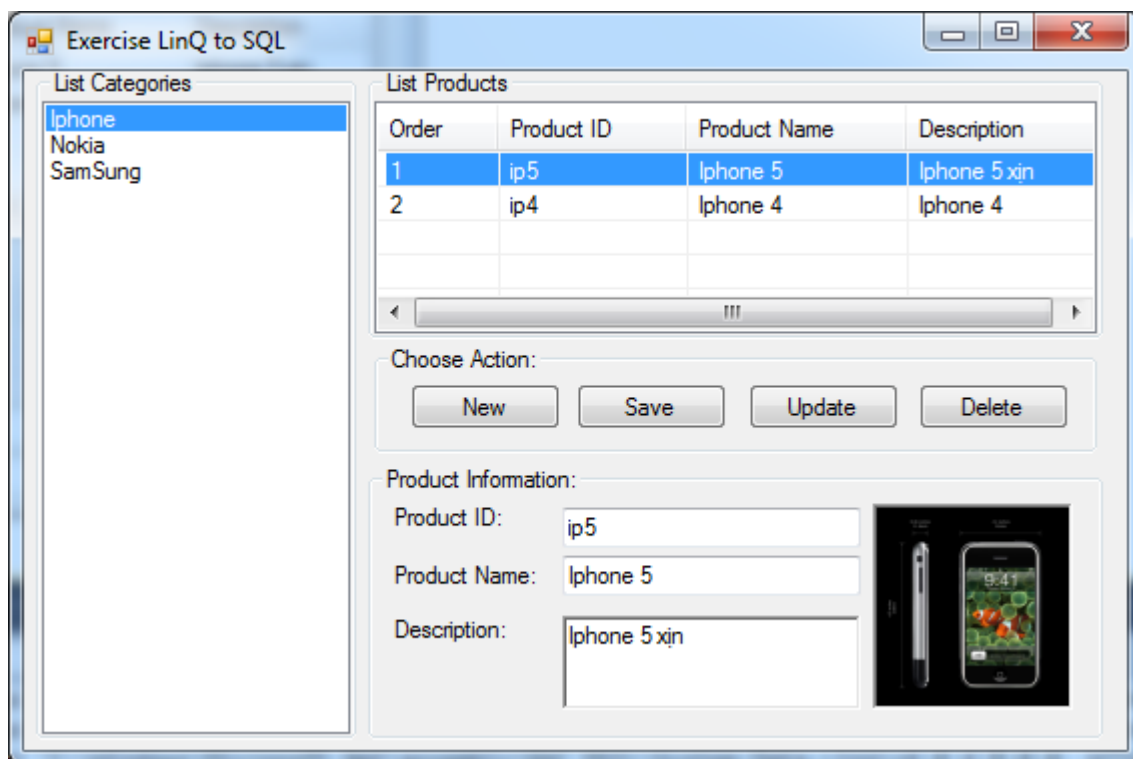
- Một số Extension Method trả về kiểu `IEnumerable<>` , ta muốn sử dụng một số hàm tiện lợi của List, Dictionary, Array... thì ta dùng các hàm ở trên: `toList`, `toArray`, `toDictionary`
- Một số trường hợp ta muốn chuyển qua Dictionary để xử lý theo Key, Value. Dictionary rất hữu dụng ở việc tối ưu tốc độ xử lý. Khi xử lý liên quan tới CSDL hay đối tượng ta thường dùng Dictionary.

Bài tập 13:**Mục đích:**

- Thực hành và hiểu được LinQ to SQL, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **Mapping object và DataContext**
 - ✓ Thực hành và hiểu được **Query**
 - ✓ Thực hành và hiểu được **Insert, update, delete**
 - ✓ Và các hàm khác...

Yêu cầu:

- Dùng LinQ để viết chương trình quản lý sản phẩm được mô tả đơn giản như sau: Danh mục sản phẩm gồm Mã và tên. Mỗi một danh mục sẽ có nhiều sản phẩm, thông tin mỗi sản phẩm bao gồm: Mã sản phẩm, tên sản phẩm, mô tả và hình ảnh.
- Thiết kế giao diện như hình bên dưới và thực hiện các yêu cầu:



- Khi khởi động chương trình sẽ tải toàn bộ danh mục sản phẩm vào listbox bên trái
- Khi người sử dụng Click chuột vào từng danh mục thì hiển thị danh sách sản phẩm của danh mục đó vào ListView.
- Khi người sử dụng Click chuột vào từng sản phẩm trong Listview thì hiển thị chi tiết thông tin sản phẩm vào phần bên dưới.
- Double click vào PictureBox để cho phép thay đổi hình ảnh sản phẩm
- Thực hiện các thao tác: New, Save, Update, Delete

Hướng dẫn:

- Dùng **MemoryStream** và **Image.FromStream** để đọc hình nhị phân từ CSDL lên giao diện

Bài tập 14:**Mục đích:**

- Thực hành và hiểu được LinQ to SQL, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **Mapping object và DataContext**
 - ✓ Thực hành và hiểu được **Query**
 - ✓ Thực hành và hiểu được **Insert, update, delete**
 - ✓ Và các hàm khác...

Yêu cầu:

- Viết chương trình quản lý sinh viên, giao diện như hình bên dưới:

- Nút “Tạo Lớp Mới” cho phép tạo lớp mới và đưa vào Adv Tree
- ListViewEx cho phép gom nhóm sinh viên theo lớp, mỗi lần click vào từng phần tử trong ListViewEx thì sẽ hiển thị thông tin chi tiết của sinh viên vào mục bên dưới
- Nút Lưu cho phép lưu thông tin sinh viên theo đúng như tên lớp đã chọn trong ComboBoxX: Cập nhật đồng thời lên ListViewEx và AdvTree

- Sinh viên tự tạo ImageList để gắn hình ảnh tương ứng với giới tính.
- Mỗi lần click chuột vào từng phần tử trong Adv Tree :
 - ✓ Nếu chọn vào Node gốc là “Khoa CNTT” thì liệt kê toàn bộ lớp và sinh viên gom nhóm theo lớp vào ListViewEx
 - ✓ Nếu chọn vào Node là Lớp nào đó thì liệt kê toàn bộ sinh viên của lớp đó vào ListViewEx mà thôi
 - ✓ Nếu chọn vào Node là sinh viên thì chỉ hiển thị thông tin của sinh viên đó vào ListViewEx mà thôi
- Nút “Xóa” cho phép xóa sinh viên hiện tại, cập nhật lại ListViewEx và Adv Tree

Hướng dẫn:

- Tạo 2 bảng dữ liệu trong SQL Server: Bảng Lớp (mã lớp, tên lớp).
- Bảng sinh viên (mã sinh viên, tên sinh viên, điện thoại, giới tính)
- Tất cả thao tác phải dùng LinQ to SQL

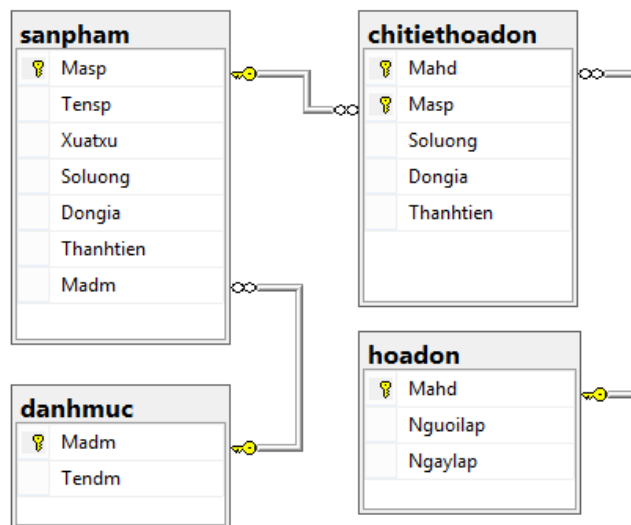
Bài tập 15:

Mục đích:

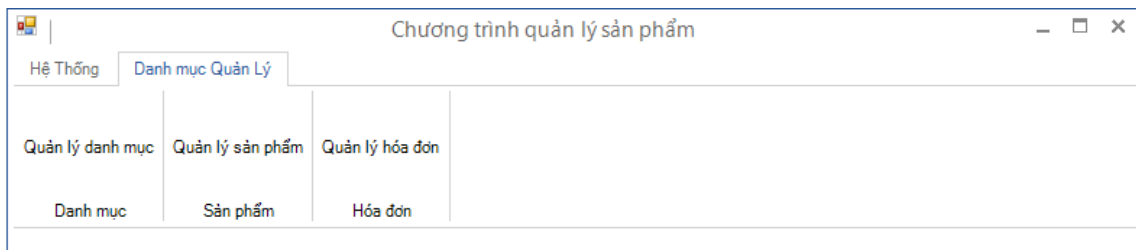
- Thực hành và hiểu được LinQ to SQL, cụ thể ở một số hàm:
 - ✓ Thực hành và hiểu được **Join**
 - ✓ Thực hành và hiểu được **Stored procedure**
 - ✓ Và các hàm khác...

Yêu cầu:

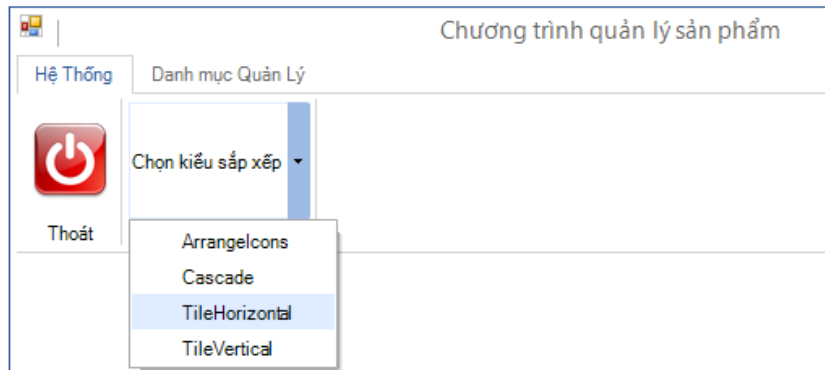
- Công ty XYZ chuyên kinh doanh trong lĩnh vực mua bán sản phẩm, trong những năm gần đây quy mô công ty ngày càng mở rộng nên ban lãnh đạo có nhu cầu viết một phần mềm quản lý sản phẩm, sau đây là một phần nhỏ của ứng dụng:



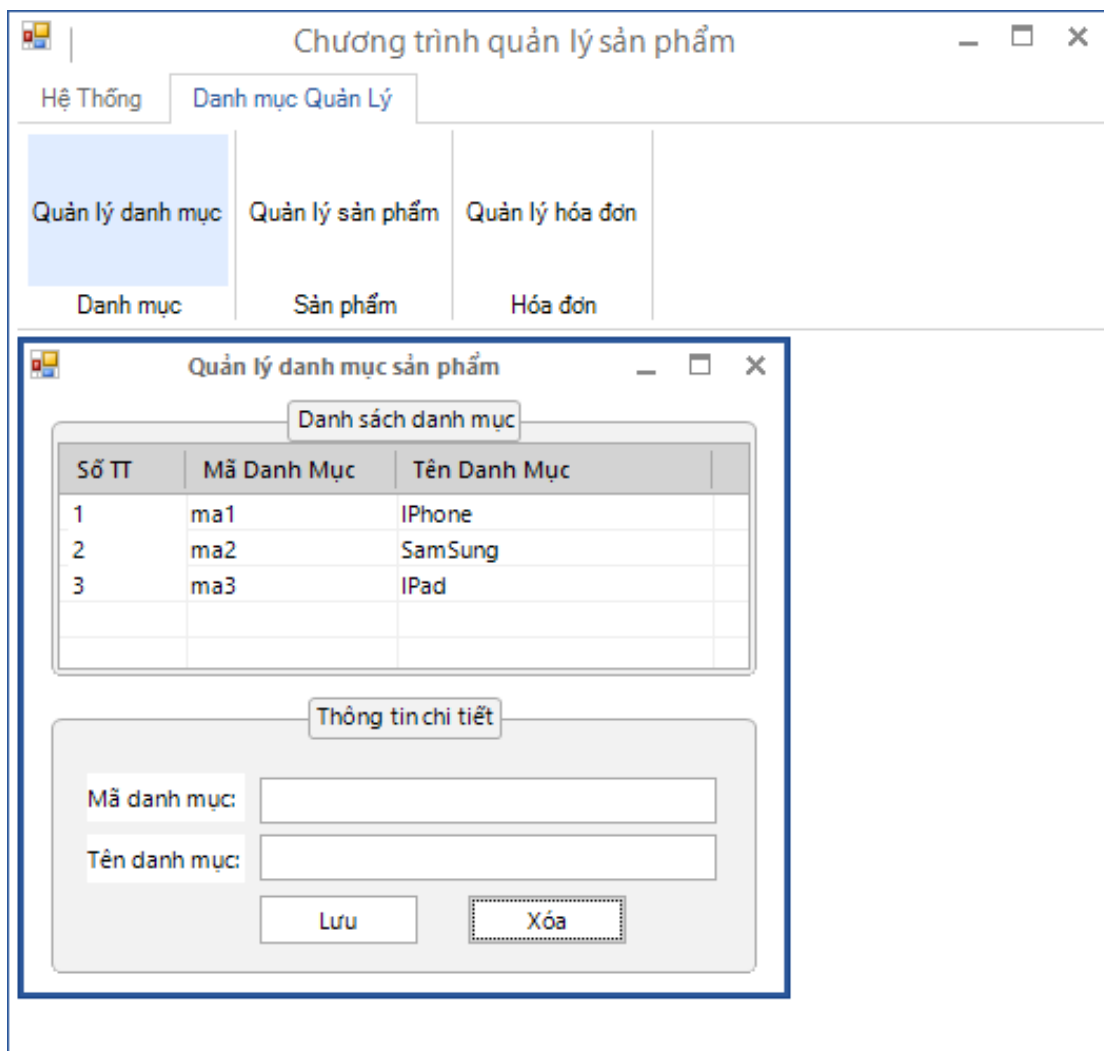
- Dưới đây là giao diện chính của chương trình:



- Tab Danh mục quản lý: Quản lý danh mục, quản lý sản phẩm, quản lý hóa đơn



- Tab Hệ Thống: Thoát, chọn kiểu sắp xếp các form con bên trong
- Khi bấm vào nút **Quản lý danh mục**:



- + Màn hình này cho phép thêm, sửa, xóa danh mục sản phẩm.
 - Bấm lưu: Nếu mã danh mục đã tồn tại thì trở thành cập nhật, còn mã danh mục chưa tồn tại thì sẽ tự động thêm mới
 - Bấm xóa: sẽ xóa danh mục hiện tại
 - Mỗi lần bấm vào từng dòng trong ListViewEx thì sẽ hiển thị thông tin chi tiết của danh mục đó vào phần thông tin chi tiết.

-Khi bấm vào nút **Quản Lý Sản Phẩm**:

- + ListBox cho phép liệt kê toàn bộ danh mục sản phẩm đã được nhập ở phần quản lý danh mục.
- + Mỗi lần chọn từng danh mục trong ListBox thì sẽ hiển thị danh sách các sản phẩm của danh mục đó vào ListViewEx bên phải màn hình
- + Mỗi lần chọn từng sản phẩm trong ListViewEx thì sẽ hiển thị thông tin chi tiết của sản phẩm đó vào mục chi tiết sản phẩm bên dưới.
- +Nút lưu: cho phép thêm mới sản phẩm nếu như mã sản phẩm chưa tồn tại, tự động cập nhật nếu như mã sản phẩm đó đã tồn tại. Chú ý là phải lưu sản phẩm vào đúng danh mục sản phẩm đang được chọn trong ListBox
- +Nút xóa : Cho phép xóa sản phẩm hiện tại đang chọn.

The screenshot shows a software application for product management. The main window has a title bar 'Chương trình quản lý sản phẩm' and three tabs: 'Hệ Thống', 'Danh mục Quản Lý', and 'Quản lý sản phẩm'. The 'Danh mục Quản Lý' tab is active, showing a 'Quản lý danh mục' section with a list of categories: 'Danh mục', 'Sản phẩm', and 'Hóa đơn'. The 'Sản phẩm' section is selected, displaying a 'Danh mục sản phẩm' list with 'iPhone', 'SamSung', and 'IPad'. The 'iPhone' category is selected, showing a 'Danh sách sản phẩm' table with columns: STT, Mã sản phẩm, Tên sản phẩm, and Thành tiền. The table contains three rows: 1 (iPhone 3, 24.0000), 2 (iPhone 4, 45.0000), and 3 (iPhone 5, 16.0000). Below the table is a 'Chi tiết sản phẩm' form with fields for 'Mã sản phẩm', 'Tên sản phẩm', 'Số lượng', 'Đơn giá', and 'Xuất xứ', along with 'Lưu' and 'Xóa' buttons.

- Khi bấm vào **Quản lý hóa đơn**:

- + Danh sách hóa đơn sẽ được tự động hiển thị vào ListBox ở màn hình bên trái
- + Bấm Tạo hóa đơn cho phép tạo hóa đơn mới theo đúng với thông số nhập vào, cập nhật lại ListBox khi tạo hóa đơn thành công.
- + Nút xóa hóa đơn: cho phép xóa hóa đơn đang chọn
- + Mỗi lần chọn hóa đơn nào thì sẽ hiện thị danh sách các chi tiết của hóa đơn vào mục ListViewEx ở bên phải màn hình.
- + Mỗi lần chọn từng dòng trong ListViewEx thì hiển thị thông tin chi tiết của từng hàng mục trong hóa đơn vào mục nhập thông tin chi tiết
- + Bấm lưu để thêm sản phẩm cho hóa đơn
- + Bấm xóa để xóa chi tiết hóa đơn đang chọn

Chương trình quản lý sản phẩm

Hệ Thống | Danh mục Quản Lý

Quản lý danh mục | Quản lý sản phẩm | Quản lý hóa đơn

Danh mục | Sản phẩm | Hóa đơn

Quản Lý Hóa Đơn

Danh sách hóa đơn

Chi tiết hóa đơn

STT	Mã SP	Tên SP	SL	ĐG	TT
1	i3	iPhone 3	2	3.0000	6.0000
2	ip5	iPhone 5	2	5000.0000	10000.0000

Nhập thông tin chi tiết

Mã sản phẩm: iPhone 3

Số lượng: 8

Đơn giá: 3.0000

Thành tiền: 24.0000

Tạo hóa đơn | Xóa hóa đơn | Lưu | Xóa

Hướng dẫn:

- Cố gắng viết các nghiệp vụ bằng Store Procedure

Bài tập 16:**Mục đích:**

- Sử dụng QuickStart configuration

Yêu cầu:

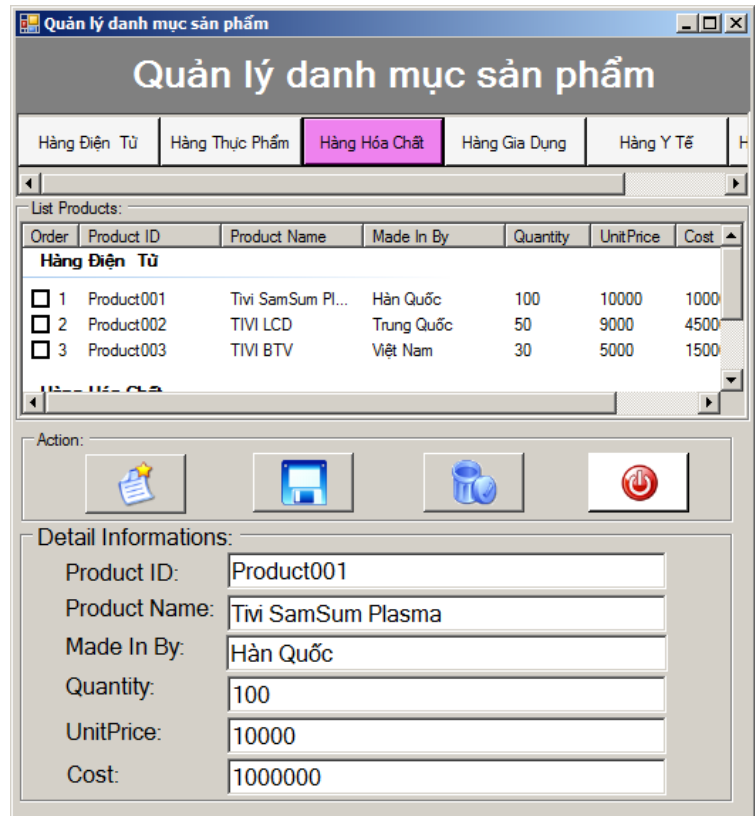
Cho cơ sở dữ liệu Sql server được mô tả như sau:

tblCatalog(CatalogID, CatalogName)

tblProduct (ProductID, MadeInBy, Quantity, UnitPrice, CatalogID)

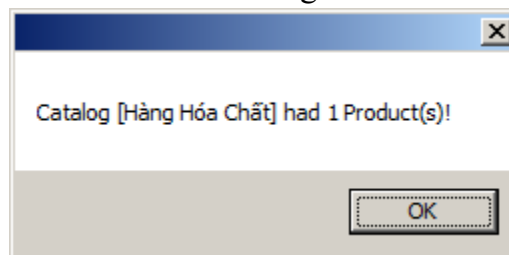
Hãy nhập một số dữ liệu mẫu và thực hiện chương trình theo yêu cầu:

1. Trong sự kiện Form Load chương trình phải tự động đọc tất cả danh sách Categories và hiển thị ra các Button như yêu cầu (viết code để load Button động), mặc định toàn bộ danh sách sản phẩm sẽ được hiển thị theo nhóm danh mục trong ListView. Viết các sự kiện Click chuột cho các Button động này (hình 1):



Hình 1

- a. Khi click chuột vào Button (theo tên danh mục) thì chương trình sẽ hiển thị số lượng sản phẩm có trong danh mục này. Ví dụ khi người sử dụng click chuột vào Button có nhãn là **Hàng Hóa Chất** thì ta có Message bên dưới:



- b. Button nào được click chuột thì Button đó sẽ có màu hồng, các Button còn lại sẽ có cùng màu xám.
- 2) Viết sự kiện cho ListView:
Mỗi lần click chuột vào từng ListViewItem thì chương trình sẽ đọc thông tin chi tiết của Product này và đưa vào mục Detail Informations, biết rằng $Cost = Quantity * UnitPrice$.
 - 3) Viết sự kiện cho các button Thêm, Lưu, Xóa, Thoát:

- Thêm mới:** Khi click chuột vào button này, chương trình chỉ cần xóa trắng toàn bộ TextBox, đồng thời cho TextBox ProductID được Focus.
- Lưu:** Cho phép lưu thông tin Product vào bảng product và cập nhật lại giao diện sử dụng.
- Xóa:** Cho phép xóa Product đang chọn và cập nhật lại giao diện sử dụng.
- Thoát:** Khi bấm vào nút này, sẽ hiển thị MessageBox hỏi người sử dụng có muốn thoát khỏi chương trình hay không, nếu có thì thoát.

*Chú ý trong phần Detail Information:

- Text Box Made In By: Sử dụng chức năng **AutoCompleteCustomSource** để hỗ trợ người sử dụng tiện lợi trong quá trình nhập liệu.
- TextBox Cost: Sẽ tự động cập nhật dữ liệu = Quantity * UnitPrice khi người sử dụng rời chuột khỏi TextBox UnitPrice.

Bài tập 17:

Yêu cầu:

Cho cơ sở dữ liệu SQL Server (**dbLibraries**) được mô tả như bên dưới, sinh viên hãy thiết kế và tạo mối quan hệ giữa các bảng dữ liệu theo đúng như yêu cầu (**1đ**).

Bảng **tblPublisher** dùng để lưu danh sách các nhà xuất bản, khóa chính là **PublisherCode**, Một nhà xuất bản sẽ có một hoặc nhiều đầu sách.

tblPublisher			
Columns Name	Data Type	Data Size	Description
<u>PublisherCode</u>	VARCHAR	50	Mã nhà xuất bản (khóa chính)
PublisherName	NVARCHAR	150	Tên nhà xuất bản
Address	NVARCHAR	150	Địa chỉ
Phone	VARCHAR	20	Điện thoại

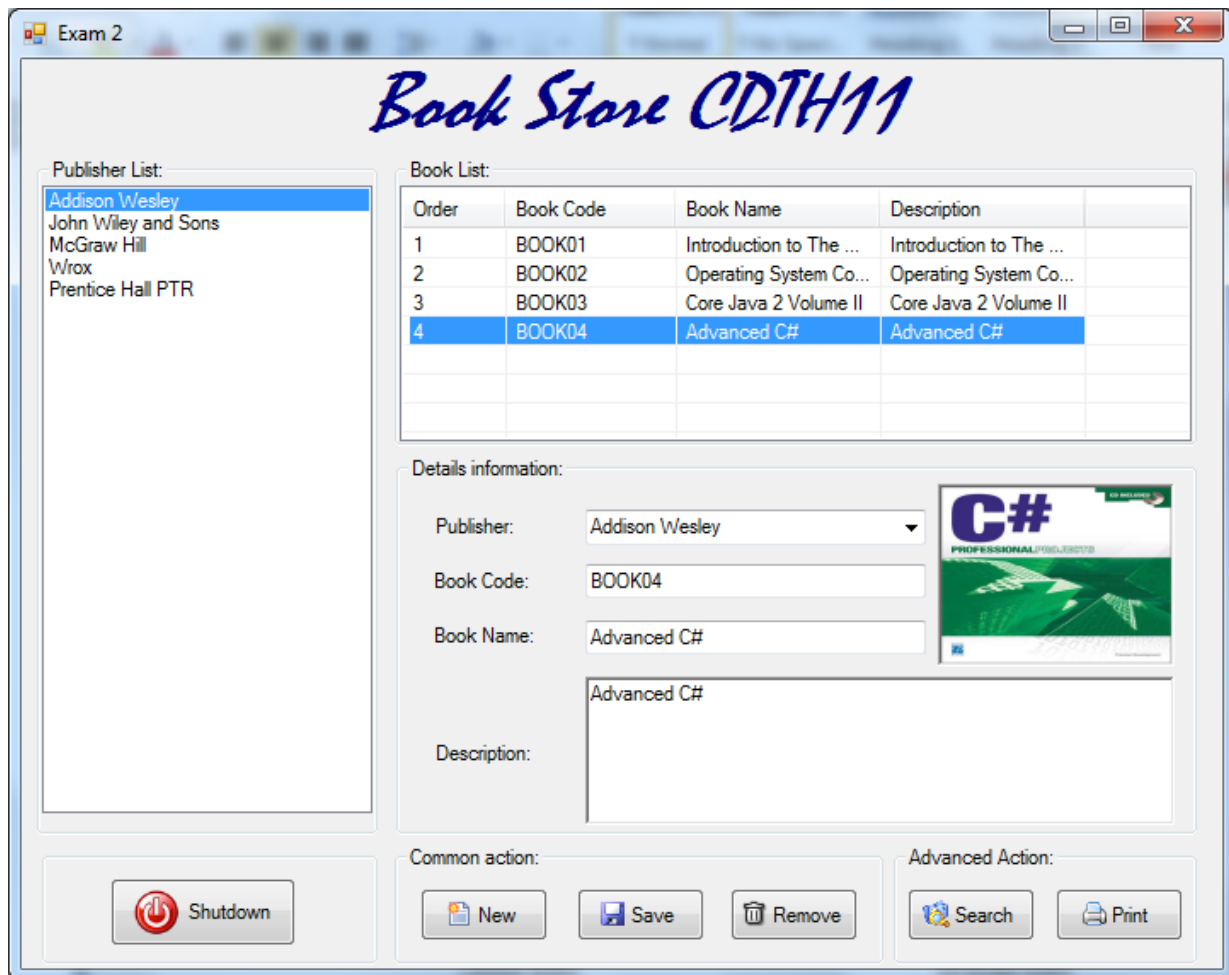
Bảng **tblBook** dùng để lưu danh sách các đầu sách, khóa chính là **BookCode**, khóa ngoại là **PublisherCode**, Khóa ngoại này sẽ tham chiếu tới cột **PublisherCode** ở trong bảng **tblPublisher**

tblBook			
Columns Name	Data Type	Data Size	Description
<u>BookCode</u>	VARCHAR	50	Mã sách (khóa chính)
BookName	NVARCHAR	150	Tên sách
BookImage	NVARCHAR	255	Tên hình
Description	NVARCHAR	MAX	Tóm tắt
<i>PublisherCode</i>	VARCHAR	50	Mã nhà xuất bản(khóa ngoại)

Chú ý: Sau khi tạo xong cơ sở dữ liệu và nhập một số dữ liệu mẫu. Sinh viên phải backup lại Cơ sở dữ liệu (có tên trùng với tên Cơ sở dữ liệu gốc), sau đó chép tập tin backup này vào thư mục bài thi.

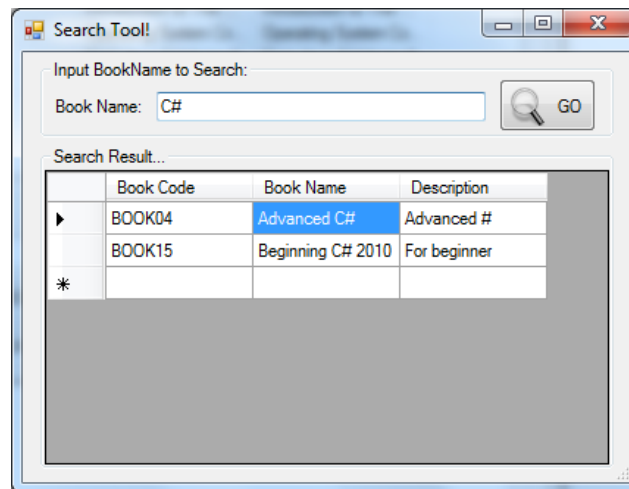
Sinh viên dùng kiến trúc đa tầng để thực hiện các yêu cầu sau:

Mặc định khi chạy chương trình lên thì giao diện chính của chương trình sẽ như hình bên dưới:



- 1) Chương trình sẽ load toàn bộ danh sách nhà xuất bản lên Listbox. Khi người sử dụng click chọn nhà xuất bản thì toàn bộ đầu sách của nhà xuất bản này sẽ được hiển thị lên Listview(3đ).
- 2) Viết sự kiện cho Listview: Khi chọn từng đầu sách trong Listview, chương trình sẽ hiển thị thông tin chi tiết từng đầu sách bao gồm cả hình ảnh (1đ).
- 3) Viết sự kiện cho các nút lệnh: New, Save, Remove, Shutdown (3đ):
 - a. **New:** Chương trình phải xóa trắng các textbox và focus tới Book Code.
 - b. **Save:** Nếu Book Code đã tồn tại thì chương trình tự động cập nhật thông tin, nếu chưa tồn tại thì chương trình tự động thêm mới. Doubleclick vào hình ảnh để thay đổi hình ảnh (dùng OpenFileDialog).
 - c. **Remove:** Chương trình sẽ xóa Book đang chọn.
 - d. **Shutdown:** Hiển thị messagebox hỏi người sử dụng có muốn thoát hay không, nếu chọn thoát thì tắt chương trình.

Chú ý: Mọi thao tác làm thay đổi cơ sở dữ liệu đều phải cập nhật lại giao diện.
- 4) Viết sự kiện cho các nút lệnh: Search, Print (2đ):
 - a. **Search:** Khi người sử dụng chọn nút Search, chương trình sẽ hiển thị giao diện bên dưới để cho phép người sử dụng nhập tên Book vào, nhấn nút GO để hiển thị kết quả tìm kiếm vào DataGridView:



- b. **Print:** Khi chọn chức năng này, chương trình sẽ in toàn bộ danh sách các đầu sách (Chú ý dùng Crystal Report).

Bài tập 18:

Yêu cầu:

- Công ty XYZ chuyên hoạt động trong lĩnh vực xuất nhập khẩu, trong những năm gần đây quy mô công ty ngày càng mở rộng vì vậy ban lãnh đạo có nhu cầu xây dựng phần mềm quản lý nhập kho. Mỗi lần có hàng nhập về, công ty cần lưu trữ vào phiếu nhập kho bao gồm thông tin Mã phiếu, ngày lập phiếu và danh sách các hàng hóa vào phiếu nhập kho.

Sau đây là một phần thiết kế cơ sở dữ liệu:

SanPham(MaSanPham, TenSanPham, DonGia, LoaiSanPham)

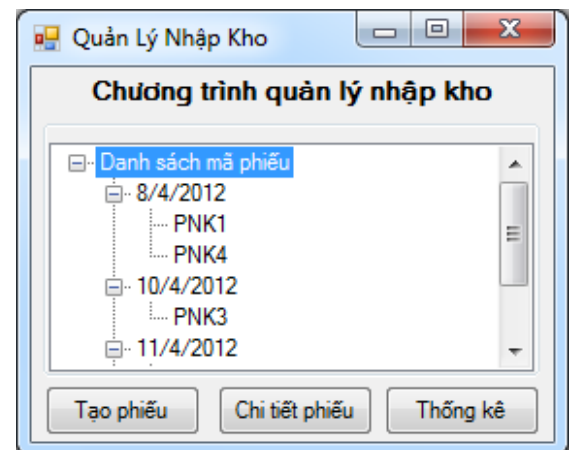
PhieuNhapKho(MaPhieu, NgayNhap)

ChiTietPhieu(MaPhieu, MaSanPham, SoLuong)

LoaiSanPham: 1 nếu là Hàng Cao Cấp, 0 nếu là hàng Thông Thường.

Câu 1 (1đ): Thiết kế cơ sở dữ liệu theo mô tả như trên (sinh viên tự chọn kiểu dữ liệu thích hợp), tạo mối quan hệ giữa các bảng. Nhập dữ liệu mẫu, mỗi bảng ít nhất 3 dòng.

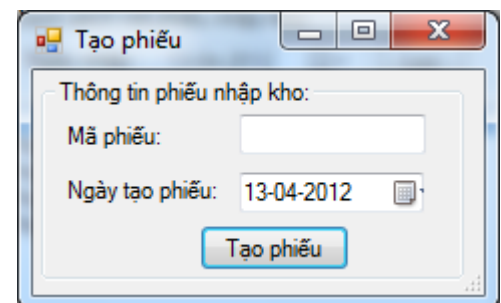
Câu 2 (3đ): Thiết kế giao diện chính của chương trình như **hình 1**. Khi màn hình chính hiện lên thì toàn bộ danh sách mã phiếu nhập kho sẽ được hiển thị vào TreeView theo ngày.



Hình 1

Câu 3 (1.5đ): Khi bấm vào nút **Tạo phiếu**, chương trình sẽ hiển thị cửa sổ như **hình 2**.

- Tại cửa sổ **Tạo phiếu nhập kho**, khi bấm nút Tạo phiếu chương trình sẽ kiểm tra mã phiếu đã tồn tại hay chưa, nếu đã tồn tại thì xuất thông báo “Mã phiếu đã tồn tại” (0.5đ). Nếu chưa tồn tại thì cho phép thêm Mã phiếu mới này vào cơ sở dữ liệu, đóng cửa sổ này đồng thời cập



Hình 2

nhập lại danh sách mã phiếu trong màn hình chính (1đ).

Câu 4 (3.5đ): Khi bấm nút **Chi tiết phiếu**, chương trình sẽ hiển thị cửa sổ Chi Tiết Phiếu Nhập của mã phiếu đang chọn trong TreeView như **hình 3**.

- Hiển thị danh sách sản phẩm của mã phiếu này vào ListView (1đ).
- Combobox Mã sản phẩm sẽ hiển thị toàn bộ mã sản phẩm từ cơ sở dữ liệu, khi chọn mã sản phẩm chương trình sẽ hiển thị thông tin chi tiết của sản phẩm vào các ô: Tên sản phẩm, Đơn giá và Loại sản phẩm (các ô này thiết lập thuộc tính chỉ đọc) (1.5đ).
- Khi chọn nút Thêm, chương trình sẽ đưa sản phẩm vào bảng ChiTietPhieu tương ứng với Mã phiếu đang chọn và Số lượng nhập, đồng thời cập nhập lên ListView (1đ).

Mã SP	Tên SP	Số lượng
SP1	Tủ lạnh Sharp	10
Sp2	Máy Giặt Sanyo	2
Sp3	Máy Điều hòa N...	5

Mã sản phẩm: Sp5

Tên sản phẩm: LabTop Sony Vaio

Đơn giá: 27

Loại sản phẩm: Hàng cao cấp

Số lượng nhập:

Thêm

Hình 3

Câu 5 (1đ): Khi chọn nút **Thống kê**, chương trình xuất thông báo mã phiếu có số lượng sản phẩm là hàng cao cấp nhiều nhất như **hình 4**.

Mã phiếu có số lượng sản phẩm là hàng cao cấp nhiều nhất:

Mã phiếu =[PNK4]

Ngày Nhập =[8-4-2012]

Tổng số lượng hàng cao cấp =[90]

OK

Hình 4